SPECIFICATION:

Page 1, please insert the following paragraphs after the paragraph in lines 21-26.

Some additional concepts will be defined next. These concepts are well known to or easily understood by the person with a regular skilled in the art. They will help to define the scope of the invention.

The expression is made up of elements (which have already been defined), OPERATORS, COMPARATORS. OPERATORS are those entities such as AND, OR, +, - and so on. COMPARATORS are those entities such as <, >, = and so on. There might exist other types of operators and comparators, because the actual choice of operators and comparators is defined by the designer of the expression and/or the designer of the actual embodiment of the invention. Comparators are the basic element used for creating predicates in boolean expressions and database queries. For example, a comparator "is" is often employed in many database system. Regardless of the actual operators and comparators that are used, both operators and comparators are concepts well known by the person skilled in the art.

An ITEM is the smallest fragment of the expression that has meaning in itself. For quantitative expressions, the items would be the variables and constants used in the expression. For example, in the formula "A + 3*(B+D*(F-H-J))", "A", "3", "B", "D", "F", "H" and "J" would be items. For logical formulas, the items would be the basic predicates, for example for "(Language=Spanish OR Cover=Hard) AND Theme=Novel AND NOT Theme=Essay", the items would be "Language=Spanish", "Cover=Hard", "Theme=Novel" and "Theme=Essay".

A SUBEXPRESSION is a fragment of the expression that has meaning in itself, is correctly formed, and is not the expression itself. For example, for the previous quantitative expression, "F-H-J" would be a subexpression, but "B+D" would not,

RECEIVED
CENTRAL FAX CENTER
JUN 10 2008

because "D" is associated to the operator and elements on its right. It must be noted that items are a type of subexpressions. In fact, items are the smallest subexpressions in an expression. It must be emphasized that in this patent application, the expression itself is not considered to be a subexpression, so all subexpressions are always smaller than the expression. In other words the expression contains all subexpressions, but no subexpression contains the expression.

An INTRODUCED SUBEXPRESSION is a subexpression to which its operators have been added. For example, "*(F-H-J)" is an introduced subexpression, and "AND NOT Theme=Essay" is an introduced subexpression. To facilitate the explanation, subexpressions which are not introduced subexpressions will also be called REGULAR SUBEXPRESSIONS.

An INTERMEDIATE SUBEXPRESSION is a subexpression that is not an item. Therefore, using the terms introduced in the previous paragraph, there can exist:

- INTERMEDIATE INTRODUCED SUBEXPRESSIONS, which are introduced subexpressions that are not items, and
- INTERMEDIATE REGULAR SUBEXPRESSIONS, which are intermediate subexpressions that are not introduced subexpressions.

Page 14. Please insert the following paragraphs after the paragraph in lines 34-35:

The graphical trees created by the current invention are characterized by the following:

- One or more of their nodes contains an intermediate subexpression. (It could also be said that "one or more of their nodes 'is' or 'shows' an intermediate subexpression") For example, in Figure 1 node 1001 is an intermediate subexpression. In fact, it is not an item, because it is composed by two items (B and C) and an operator +. Node 1002 is an item. .

- One or more of the nodes contains an introduced intermediate subexpression. Node 1003 is an introduced intermediate subexpression, because it contains the operator 1004. As mentioned before, some times an introduced subexpression might have two operators, such as "AND NOT", in logical expressions.
- One more of the nodes contains an regular intermediate subexpression. Node 1001 is a regular subexpression.

The escalator structure also shares these characteristics. For example, Figure 11 shows that a given subexpression can be readily seen by navigating down the structure and up again. For example, the subexpression "A + (B+D)/F" can be observed by starting in element 1101, moving down one level, and continuing at that level to reach element F, where we can move up again to the starting level.

This can be more easily seen by using different features of the invention, such as emphasizing. Figure 25, 26 and 27 illustrate how this function allows to more easily see the different subexpressions. Collapsing/expanding nodes in the escalator structure also serve the same purpose, as shown in Figure 17 and Figure 18.